**DB Input Format**

* For this code to execute successfully, ensure Hadoop version you are using is 1.2.1 and above.
* Also MySQL Connector jar **“mysql-connector-java-5.1.34.jar”** is included in the project.
* Same needs to be copied to lib folder, other it will throw runtime error.

In case if you are using Hadoop 2..make sure it is copied at **“hadoop->share->hadoop>mapreduce>lib”**

**Code as follows :**

**DBInputWritable.java**

-----------------------------

import java.io.DataInput;

import java.io.DataOutput;

import java.io.IOException;

import java.sql.ResultSet;

import java.sql.PreparedStatement;

import java.sql.SQLException;

import org.apache.hadoop.io.Writable;

import org.apache.hadoop.mapreduce.lib.db.DBWritable;

public class DBInputWritable implements Writable, DBWritable

{

private int id;

private String name;

public void readFields(DataInput in) throws IOException { }

public void readFields(ResultSet rs) throws SQLException

//Resultset object represents the data returned from a SQL statement

{

id = rs.getInt(1);

name = rs.getString(2);

}

public void write(DataOutput out) throws IOException { }

public void write(PreparedStatement ps) throws SQLException

{

ps.setInt(1, id);

ps.setString(2, name);

}

public int getId()

{

return id;

}

public String getName()

{

return name;

}

}

-------------------------

**DBOutputWritable.java**

import java.io.DataInput;

import java.io.DataOutput;

import java.io.IOException;

import java.sql.ResultSet;

import java.sql.PreparedStatement;

import java.sql.SQLException;

import org.apache.hadoop.io.Writable;

import org.apache.hadoop.mapreduce.lib.db.DBWritable;

public class DBOutputWritable implements Writable, DBWritable

{

private String name;

private int count;

public DBOutputWritable(String name, int count)

{

[this.name](http://this.name/) = name;

this.count = count;

}

public void readFields(DataInput in) throws IOException { }

public void readFields(ResultSet rs) throws SQLException

{

name = rs.getString(1);

count = rs.getInt(2);

}

public void write(DataOutput out) throws IOException { }

public void write(PreparedStatement ps) throws SQLException

{

ps.setString(1, name);

ps.setInt(2, count);

}

}

-------------------------

**Main.java**

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.lib.db.DBConfiguration;

import org.apache.hadoop.mapreduce.lib.db.DBInputFormat;

import org.apache.hadoop.mapreduce.lib.db.DBOutputFormat;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.NullWritable;

public class Main

{

public static void main(String[] args) throws Exception

{

Configuration conf = new Configuration();

DBConfiguration.configureDB(conf,

"com.mysql.jdbc.Driver", // driver class

"jdbc:mysql://localhost:3306/testDb", // db url

"root", // user name

"root"); //password

//Check username and password, database name, table name in your case

Job job = new Job(conf);

job.setJarByClass(Main.class);

job.setMapperClass(Map.class);

job.setReducerClass(Reduce.class);

job.setMapOutputKeyClass(Text.class);

job.setMapOutputValueClass(IntWritable.class);

job.setOutputKeyClass(DBOutputWritable.class);

job.setOutputValueClass(NullWritable.class);

job.setInputFormatClass(DBInputFormat.class);

job.setOutputFormatClass(DBOutputFormat.class);

DBInputFormat.setInput(job, DBInputWritable.class,

"studentinfo", //input table name

null,

null,

new String[] { "id", "name" } // table columns

);

DBOutputFormat.setOutput(

job,

"output", // output table name

new String[] { "name", "count" } //table columns

);

System.exit(job.waitForCompletion(true) ? 0 : 1);

}

}

----------------------

**Map.java**

import java.io.IOException;

import org.apache.hadoop.mapreduce.Mapper;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.io.IntWritable;

public class Map extends Mapper<LongWritable, DBInputWritable, Text, IntWritable>

{

private IntWritable one = new IntWritable(1);

protected void map(LongWritable id, DBInputWritable value, Context ctx)

{

try

{

String[] keys = value.getName().split(" ");

for(String key : keys)

{

ctx.write(new Text(key),one);

}

} catch(IOException e)

{

e.printStackTrace();

} catch(InterruptedException e)

{

e.printStackTrace();

}

}

}

---------------------

**Reduce.java**

1, shrikant

1, China

import java.io.IOException;

import org.apache.hadoop.mapreduce.Reducer;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.NullWritable;

public class Reduce extends Reducer<Text, IntWritable, DBOutputWritable, NullWritable>

{

protected void reduce(Text key, Iterable<IntWritable> values, Context ctx)

{

int sum = 0;

for(IntWritable value : values)

{

sum += value.get();

}

try

{

ctx.write(new DBOutputWritable(key.toString(), sum), NullWritable.get());

} catch(IOException e)

{

e.printStackTrace();

} catch(InterruptedException e)

{

e.printStackTrace();

}

}

}